

11/27/00
JC836 U.S. PTO

U.S. PTO
09/24/337
11/27/00

PATENT APPLICATION
Express Mail Label No. E1436467949US
Attorney Docket No. NA00-13402

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

UTILITY PATENT
APPLICATION TRANSMITTAL LETTER

Asst. Commissioner for Patents
Box Patent Application
Washington, D.C. 20231

Sir:

Enclosed for filing is an ☒ original patent application or, ☐ a continuation-in-part patent application, by inventor(s) William F. Price III, entitled METHOD AND APPARATUS FOR MANAGING PUBLIC KEYS THROUGH A SERVER.

No. of pages in Application: 24; No. of Claims: 27.

No. of Sheets of Drawings: Formal: 5, Informal: 0.

Also enclosed are:

- ☐ a claim for foreign priority under 35 U.S.C. §§ 119 and/or 365 in
☐ a separate document ☐ the declaration;
- ☐ a certified copy of the priority document;
- ☐ an Associate Power of Attorney;
- ☐ ___ verified statement(s) claiming small entity status;
- ☒ a Combined Declaration and Power of Attorney of the inventor(s);
- ☐ a signed Combined Declaration and Power of Attorney of the inventors will follow;
- ☒ an Assignment document and form PTO-1595;
- ☐ a Power of Attorney by Assignee; and
- ☐ Information Disclosure Statement and Form PTO-1449.

The fee has been calculated as follows:

CLAIMS					
	NO. OF CLAIMS		EXTRA CLAIMS	RATE	FEE
Basic Application Fee					\$710.00
Total Claims	27	MINUS 20 =	7	\$18.00=	\$126.00
Independent Claims	3	MINUS 3 =	0	\$80.00=	\$0.00
If multiple dependent claims are presented, add \$260.00					0
Total Application Fee					\$836.00
If verified statement claiming small entity status is enclosed, subtract 50% of Total Application Fee					
Add Recording Fee of \$40.00 if Assignment document is enclosed					\$40.00
TOTAL APPLICATION FEE DUE					\$876.00

- ☒ A check in the amount of \$ 876.00 is enclosed.
- ☐ Application fee will follow with missing parts.
- ☒ Please deduct any underpayments, credit any overpayments, and charge all required extension of time fees to Deposit Account Number 50-1003.

Please direct all correspondence concerning the above-identified application to the following address:

A. Richard Park
Park & Vaughan LLP
508 Second Street, Suite 201
Davis, CA 95616
(530) 759-1661



22835
PATENT TRADEMARK OFFICE

Respectfully submitted,

By A. Richard Park
A. Richard Park
Registration No. 41,241

Date: November 27, 2000

**PATENT APPLICATION
ATTORNEY DOCKET NO. NA00-13402**

5

10 **METHOD AND APPARATUS FOR MANAGING
PUBLIC KEYS THROUGH A SERVER**

Inventor: William F. Price III

15 **Related Application**

The application hereby claims priority under 35 U.S.C. § 119 to
Provisional Patent Application No. 60/230,235 filed on September 1, 2000.

The subject matter of this application is related to the subject matter in a
co-pending non-provisional application by the same inventor as the instant
20 application and filed on the same day as the instant application entitled, "Method
and Apparatus for Periodically Removing Invalid Public Keys from a Public Key
Server," having serial number TO BE ASSIGNED, and filing date TO BE
ASSIGNED (Attorney Docket No. NA00-13801).

25 **BACKGROUND**

Field of the Invention

The present invention relates to the field of computer security. More
specifically, the present invention relates to a method and an apparatus for

managing public keys by using a server that stores associations between public keys and email addresses.

Related Art

5 The advent of computer networks has led to an explosion in the development of applications that facilitate rapid dissemination of information. In particular, electronic mail (email) is becoming the predominant method for communicating textual and other non-voice information. Using email, it is just as easy to send a message to a recipient on another continent as it is to send a
10 message to a recipient within the same building. Furthermore, an email message typically takes only minutes to arrive, instead of the days it takes for conventional mail to snake its way along roads and through airports.

 One problem with email is that it is hard to ensure that sensitive information sent through email is kept confidential. This is because an email
15 message can potentially traverse many different computer networks and many different computer systems before it arrives at its ultimate destination. An adversary can potentially intercept an email message at any of these intermediate points along the way.

 One way to remedy this problem is to “encrypt” sensitive data using an
20 encryption key so that only someone who possesses a corresponding decryption key can decrypt the message. (Note that for commonly used symmetric encryption mechanisms the encryption key and the decryption key are the same key.) A person sending sensitive data through email can encrypt the sensitive data using the encryption key before it is sent through email. At the other end, the
25 recipient of the email can use the corresponding decryption key to decrypt the sensitive information.

Managing encryption keys for the millions of users who can potentially send encrypted email message is a challenging task. Some existing key management systems operate by enrolling public keys for users with an "identity authority." An identity authority typically operates by verifying the identities of owners of public keys as well as keeping track of revoked public keys.

However, existing systems have a number of shortcomings. The verification process is often cumbersome. It typically involves some type of manual check, such as making a telephone call, taking a fingerprint, or receiving personal information from an owner of a public key. Although such manual checks provide a measure of security, they are time-consuming and can be impractical to perform for a large number of users.

Another shortcoming is that the key revocation process does not work well. Some existing systems make use of a "certificate revocation list" (CRL), which contains a listing of revoked certificates. Before using a public key, a client typically checks a locally stored copy of a CRL to verify that the public key has not been revoked. However, a locally stored copy of a CRL may be updated only occasionally (for example, once a week), which means the locally stored copy of the CRL may not be current. This can create problems. For example, an employee who leaves a company may continue to receive sensitive encrypted email messages until the locally stored copy of the CRL is updated.

Furthermore, a CRL can grow very large over time as more and more certificates are revoked. In some cases, a CRL can contain millions of entries! Hence, a locally stored copy of a CRL can require a large amount space to store, and can be cumbersome to update.

What is needed is a method and apparatus for managing encryption keys that does not require a time-consuming manual check during the verification

process, and that does not suffer from the shortcomings of using a CRL to keep track of revoked keys.

SUMMARY

5 One embodiment of the present invention provides a system for managing public keys through a server that stores associations between public keys and email addresses. This system operates by receiving a first message from a client containing a request for approval of a client public key along with the client public key. In response this request for approval, the system sends a second
10 message to the client containing a request for identity confirmation that includes the client public key. If a third message is received from the client containing an affirmative response to the request for identity confirmation, the system stores an association between a client email address and the client public key in a database. This allows other clients to look up the client public key in the database.

15 In one embodiment of the present invention, the system additionally receives a communication from a second client that includes the client email address. In response to this communication, the system performs a lookup in the database based on the client email address to determine if the client email address is associated with the client public key. If the lookup indicates that the client
20 email address is associated with the client public key, the system sends a key identifier for the client public key from the server to the client. This key identifier allows the client to determine whether the client possesses the client public key.

 In one embodiment of the present invention, the request for approval includes key reconstitution information that allows the client to decrypt to an
25 encrypted client private key at the client if the client forgets a passphrase for decrypting the encrypted client private key. This key reconstitution information is stored in the database at the server. Note that a “passphrase” is a generalization of

a password that can potentially contain an entire phrase instead of a single password.

In one embodiment of the present invention, the system decrypts the request for approval using a server private key, and then uses the client public key
5 to verify that the request for approval is signed by a corresponding client private key.

In one embodiment of the present invention, prior to sending the second message, the system determines if the database already contains a prior client public key associated with the client email address. If so, the system includes the
10 prior client public key in the request for identity confirmation sent to the client. This enables the client to subsequently indicate that the server should replace the prior client public key with the client public key.

In one embodiment of the present invention, the system additionally receives a request at the server to remove the client public key from the database.
15 If this request is signed with a corresponding client private key, the system removes the client public key from the database.

In one embodiment of the present invention, the database contains at most one key for each email address.

In one embodiment of the present invention, the database contains at most
20 one email address for each key.

In one embodiment of the present invention, the system periodically sends a verification request from the server to the client email address asking if the client public key remains valid. If an affirmative response to the verification request is not received, the system removes the client public key from the
25 database.

One embodiment of the present invention provides a system for managing public keys through a server that stores associations between public keys and

email addresses. This system operates by receiving a client public key from a client, and then storing the client public key in a database at the server. The system then allows other clients to lookup the client public key in the database. The system also periodically sends a verification request from the server to the client asking if the client public key remains valid. If an affirmative response to the verification request is not received, the system removes the client public key from the database.

In one embodiment of the present invention, the system stores the client public key in the database by signing the client public key using a server private key, and then storing the signed client public key in the database.

In one embodiment of the present invention, the client public key is removed from the database only if an affirmative response is not received after sending multiple verification requests at different times.

Note that the present invention facilitates automated identity confirmation without requiring a time-consuming manual check during the verification process. Furthermore, the present invention does not rely on locally stored key management information, which can potentially be outdated, and does not suffer from having to maintain a lengthy revocation list.

BRIEF DESCRIPTION OF THE FIGURES

FIG. 1 illustrates a distributed computer system in accordance with an embodiment of the present invention.

FIG. 2 illustrates messages sent during the key enrollment process in accordance with an embodiment of the present invention.

FIG. 3 is a flow chart of the key enrollment process in accordance with an embodiment of the present invention.

FIG. 4 is a flow chart of a client-initiated key removal process in accordance with an embodiment of the present invention.

FIG. 5 is a flow chart of the key lookup process in accordance with an embodiment of the present invention.

5 FIG. 6 is a flow chart of a server-initiated key removal process in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION

10 The following description is presented to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications to the disclosed embodiments will be readily apparent to those skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the present invention. Thus, the
15 present invention is not intended to be limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features disclosed herein.

The data structures and code described in this detailed description are typically stored on a computer readable storage medium, which may be any device
20 or medium that can store code and/or data for use by a computer system. This includes, but is not limited to, magnetic and optical storage devices such as disk drives, magnetic tape, CDs (compact discs) and DVDs (digital versatile discs or digital video discs), and computer instruction signals embodied in a transmission medium (with or without a carrier wave upon which the signals are modulated).
25 For example, the transmission medium may include a communications network, such as the Internet.

Networked Computer System

FIG. 1 illustrates a networked computer system 100 in accordance with an embodiment of the present invention. Networked computer system 100 includes clients 108 and 118, which are coupled to server 120 through network 110.

5 Clients 108 and 118 can generally include any type of computer system, including, but not limited to, a computer system based upon a microprocessor, a mainframe processor, a device controller, and a computational engine within an appliance. In the embodiment illustrated in FIG. 1, client 108 contains private key 104 and public key 106, which collectively form a private key-public key pair in such a way that a message to be encrypted using public key 106 and decrypted using
10 private key 104. Note that private key 104 cannot be deduced from public key 106 in a tractable amount of computational time. Similarly, client 118 contains private key 114 and public key 116, and server 120 includes private key 124 and public key 126.

15 Network 110 can include any type of wire or wireless communication channel capable of coupling together client 108, client 118 and server 120. This includes, but is not limited to, a local area network, a wide area network, or a combination of networks. In one embodiment of the present invention, network 110 includes the Internet.

20 Server 120 can include any node on a computer network including a mechanism for servicing requests from clients 108 and 118 for computational and/or data storage resources. In the embodiment of the present invention illustrated in FIG. 1, server 120 processes requests to enroll keys in database 122 and to lookup keys in database 122. Database 122 can include any type of storage
25 system that is capable of storing data for server 120. In one embodiment of the present invention, database 122 stores associations between email addresses and public keys. Also note that in one embodiment of the present invention database

122 operates in conformance with the lightweight directory access protocol (LDAP).

Key Enrollment Process

5 FIG. 2 illustrates messages sent between client 108 and server 120 during the key enrollment process in accordance with an embodiment of the present invention. In order to enroll a public key, client 108 sends a message to server 120 containing a request for approval 202. Request for approval 202 contains a message type field 203, which identifies the type of message. In this case,
10 message type field 203 specifies that the message is a “request for approval.” Request for approval 202 also includes a public key 106 belonging to client 108. Note that client 108 holds a corresponding private key 104.

Request for approval 202 optionally contains key reconstitution information 204, which can be used to decrypt private key 104 if user 101 forgets the passphrase
15 that was used to encrypt private key 104. For example, this key reconstitution information 204 may allow user 101 to decrypt private key 104 by remembering answers to three of five questions, such as “what is your mother’s maiden name?”, or “what was your first dog’s name?”. This key reconstitution information is discussed in more detail in a pending U.S. patent application entitled, “Method and
20 Apparatus for Reconstituting an Encryption Key Based on Multiple User Responses,” by inventor William F. Price III, Serial No. 09/429,217, filed October 28, 1999. This application is hereby incorporated by reference to describe the key reconstitution process.

In one embodiment of the present invention, request for approval 202 is
25 encrypted with public key 126 belonging to server 120, so that request for approval 202 can only be decrypted using corresponding private key 124, which is held by server 120. Request for approval 202 is also signed with private key 104

belonging to client 108. This enables server 120 to use corresponding public key 106 to verify that client 108 signed request for approval 202.

Upon receiving request for approval 202, server 120 sends a message to the email address of client 108. This message includes a request for ID confirmation 206. Request for ID confirmation 206 contains a message type field 207, which specifies that the message is a “request for ID confirmation.” Request for ID confirmation 206 also contains public key 106, as well as existence flag 208 and possibly an old public key 210. If old public key 210 already exists for client 108 in database 122, server 120 sets existence flag 208 to TRUE, and includes old public key 210 in request for ID confirmation 206.

In one embodiment of the present invention, request for ID confirmation 206 is encrypted with public key 106 belonging to client 108, so that request for ID confirmation 206 can only be decrypted using corresponding private key 104, which is held by client 108. Request for ID confirmation 206 is also signed with private key 124 belonging to server 120. This enables client 108 to use corresponding public key 126 to verify that server 120 signed request for ID confirmation 206.

Upon receiving request for ID confirmation 206, client 108 sends a message to server 120, which includes response to ID confirmation 212. Response to ID confirmation 212 contains a message type field 213, which specifies that the message is a “response to ID confirmation.” Response to ID confirmation 212 also includes replacement flag 214 and public key 106. Replacement flag 214 is set to TRUE if client 108 would like to replace old public key 210 with new public key 106.

In one embodiment of the present invention, response to ID confirmation 212 is encrypted with public key 126 belonging to server 120, so that request for ID confirmation 206 can only be decrypted using corresponding private key 124,

which is held by server 120. Request for ID confirmation 206 is also signed with private key 104 belonging to client 108. This enables server 120 to use corresponding public key 106 to verify that client 108 signed response to ID confirmation 212.

5 Note that in one embodiment of the present invention, messages 202, 206 and 212 are email messages. However, note that it is sufficient for only message 206 from server 120 to client 108 to be an email message, so that server 120 can verify that the email address of client 108 is associated with public key 106.

FIG. 3 is a flow chart of the key enrollment process in accordance with an
10 embodiment of the present invention. User 101 first enters a name, an email address and a passphrase into client 108 (step 302). User 101 can optionally enter five questions and five responses to create key reconstitution information 204 as is discussed above with reference to FIG. 2 (step 304). Next, client 108 generates a key pair including private key 104 and public key 106 (step 306). Client 108 then
15 constructs request for approval 202, as is described above with reference to FIG. 2, and sends request for approval 202 to server 120 (step 308).

Upon receiving request for approval 202 (step 310), server 120 decrypts request for approval 202 using server private key 124. Server 102 also uses public key 106 to verify that request for approval 202 has been signed by private key 104
20 held by client 108. If request for approval 202 includes key reconstitution information 204, server 120 stores key reconstitution information 204 in database 122 (step 311).

Server 120 also attempts to validate that public key 106 is associated with an email address of client 108. This is done by sending a message to the email
25 address of client 108, and then receiving a response to the message.

More specifically, server 120 constructs request for ID confirmation 206, as is described above with reference to FIG. 2, and sends request for ID

confirmation 206 to client 108 (step 312). Note that if an old public key 210 is associated with the email address of client 108 in database 122, server 120 sets existence flag 208 to TRUE, and includes old public key 210 in request for ID confirmation 206.

- 5 Upon receiving request for ID confirmation 206 (step 316), client 108 decrypts request for ID confirmation using client private key 104. Client 108 also uses public key 126 to verify that request for ID confirmation 206 has been signed by private key 124 held by server 120. Next, client 108 constructs response to ID confirmation 212, as is described above with reference to FIG. 2, and sends
- 10 response to ID confirmation 212 to server 120 (step 318). Note that if client 108 would like to replace old public key 210 with new public key 106, client 108 sets replacement flag 214 within response to ID confirmation 212 to TRUE.

- Upon receiving response to ID confirmation 212, server 120 decrypts response to ID confirmation using server private key 124. Server 120 also uses
- 15 public key 106 to verify that response to ID confirmation 212 has been signed by private key 104 held client 108.

- Next, server 120 stores an association between the email address of client 108 and client public key 106 in database 122 (step 320). In one embodiment of the present invention, storing this association involves signing client public key
- 20 106 using server private key 124, and then storing the signed client public key 106 in database 122 indexed by the email address of client 108.

- Note that in one embodiment of the present invention, database 122 contains at most one public key for each email address. Furthermore, database 122 contains at most one email address for each public key.

- 25 Finally, after sending response to ID confirmation 212, client 108 periodically polls server 120 to verify that new client public key 106 has been

stored in database 122 (step 322). The enrollment process for public key 106 is now complete.

Client-Initiated Key Removal Process

5 FIG. 4 is a flow chart of a client-initiated key removal process in accordance with an embodiment of the present invention. During this process, client 108 receives a command from user 101 to unenroll client public key 106 (step 402). In response this command, client 108 constructs a removal request and sends it the server 120 (step 404). Note that constructing this removal request
10 involves signing the removal request with private key 104 belonging to client 108.

 Upon receiving the removal request (step 406), server 120 uses client public key 106 to verify that the request was validly signed with corresponding client private key 104. If the request is validly signed, server 120 removes client public key 106 from database 122.

15

Key Lookup Process

 FIG. 5 is a flow chart of the key lookup process in accordance with an embodiment of the present invention. Note that every time a client, such as client 118, sends an encrypted email message, client 118 looks up the public keys of all
20 recipients of the email message in database 122, so that client 118 can encrypt the email message to each of the recipients. Note that an encrypted email message is typically encrypted with a randomly generated session key, and this randomly generated session key is typically encrypted with the public keys of each of the recipients to form a set of encrypted session keys. This set of encrypted session
25 keys is sent along the with encrypted email message so that each of the recipients can decrypt the session key in order to decrypt the encrypted email message.

004260
In order to lookup a public key, client 118 sends a lookup request including an email address to database 122. Upon receiving the lookup request (step 502), database 122 decrypts and verifies the lookup request if necessary, and then performs a lookup using the email address (step 504). If the email address is associated with a public key 106 in database 122, database 122 returns an identifier for public key 106 (possibly signed and encrypted) to client 118 (step 506). This identifier may be a hash (message digest) created from public key 106. Note that sending the identifier can be easier than sending the public key, because this identifier is smaller than the public key. However, note that if client 118 does not possess a local copy of public key 106, database 122 eventually sends public key 106 to client 118.

Also note that in one embodiment of the present invention, communications between client 118 and database 122 involve fast LDAP communications instead of slower email messages. This makes it practical to perform lookups each time an encrypted email message is sent.

In one embodiment of the present invention, in order to reduce the number of lookups, the system does not perform lookups into database 122 every time an encrypted email message is sent, but instead uses locally stored public keys for the message recipients. These locally stored public keys are periodically updated by performing lookups into database 122.

Server-Initiated Key Removal Process

FIG. 6 is a flow chart of a server-initiated key removal process in accordance with an embodiment of the present invention. In this embodiment, server 120, periodically verifies that each public key within database 122 remains valid. If a given key is not valid, it is removed from database 122.

More specifically, for each public key stored in database 122, server 102 sends a verification request to the associated email address (step 602). This verification request includes the client public key 106, and is possibly encrypted and signed by server 120.

- 5 If an affirmative response is not received to the verification request, server 120 removes client public key 106 from database 122 (step 604).

For example, each client public key can be verified once every six months, and the verification request can be resent to client 108 every week for a month before removing public key 106 from database 122 for lack of response.

- 10 Note that the above-described process removes public keys belonging to users who lose access the their email account, or users who die. Moreover, the above-described process solves the problem of users not being able to remove their public keys if they forget their password.

- 15 The foregoing descriptions of embodiments of the present invention have been presented for purposes of illustration and description only. They are not intended to be exhaustive or to limit the present invention to the forms disclosed. Accordingly, many modifications and variations will be apparent to practitioners skilled in the art. Additionally, the above disclosure is not intended to limit the present invention. The scope of the present invention is defined by the appended
20 claims.

What Is Claimed Is:

1 1. A method for managing public keys through a server that stores
2 associations between public keys and email addresses, comprising:
3 receiving a first message from a client at the server, the first message
4 containing a request for approval of a client public key along with the client
5 public key;
6 sending a second message from the server to the client, the second
7 message containing a request for identity confirmation that includes the client
8 public key; and
9 if a third message is received from the client at the server containing an
10 affirmative response to the request for identity confirmation, storing an
11 association between a client email address and the client public key in a database,
12 so that other clients can look up the client public key in the database.

1 2. The method of claim 1, further comprising:
2 receiving a communication from a second client at the server, the
3 communication including the client email address;
4 performing a lookup in the database based on the client email address to
5 determine if the client email address is associated with the client public key;
6 if the lookup indicates that the client email address is associated with the
7 client public key, sending a key identifier for the client public key from the server
8 to the client, wherein the key identifier allows the client to determine whether the
9 client possesses the client public key.

1 3. The method of claim 1,

2 wherein the request for approval includes key reconstitution information
3 that allows the client to decrypt to an encrypted client private key at the client if
4 the client forgets a passphrase for decrypting the encrypted client private key; and
5 wherein the method further comprises storing the key reconstitution
6 information in the database.

1 4. The method of claim 1, further comprising:
2 decrypting the request for approval at the server using a server private key,
3 the request for approval having been encrypted with a corresponding server public
4 key by the client; and
5 using the client public key to verify that the request for approval is signed
6 by a corresponding client private key.

1 5. The method of claim 1, wherein prior to sending the second
2 message, the method further comprises:
3 determining if the database already contains a prior client public key
4 associated with the client email address; and
5 if the database already contains the prior client public key, including the
6 prior client public key in the request for identity confirmation sent to the client in
7 the second message, so that the client can indicate that the server should replace
8 the prior client public key with the client public key.

1 6. The method of claim 1, further comprising:
2 receiving a request at the server to remove the client public key from the
3 database;
4 if the request is signed with a corresponding client private key, removing
5 the client public key from the database.

1 7. The method of claim 1, wherein the database contains at most one
2 key for each email address.

1 8. The method of claim 1, wherein the database contains at most one
2 email address for each key.

1 9. The method of claim 1, further comprising:
2 periodically sending a verification request from the server to the client
3 email address asking if the client public key remains valid; and
4 if an affirmative response to the verification request is not received,
5 removing the client public key from the database.

1 10. A computer-readable storage medium storing instructions that
2 when executed by a computer cause the computer to perform a method for
3 managing public keys through a server that stores associations between public
4 keys and email addresses, the method comprising:
5 receiving a first message from a client at the server, the first message
6 containing a request for approval of a client public key along with the client
7 public key;
8 sending a second message from the server to the client, the second
9 message containing a request for identity confirmation that includes the client
10 public key; and
11 if a third message is received from the client at the server containing an
12 affirmative response to the request for identity confirmation, storing an
13 association between a client email address and the client public key in a database,
14 so that other clients can look up the client public key in the database.

1 11. The computer-readable storage medium of claim 10, wherein the
2 method further comprises:
3 receiving a communication from a second client at the server, the
4 communication including the client email address;
5 performing a lookup in the database based on the client email address to
6 determine if the client email address is associated with the client public key;
7 if the lookup indicates that the client email address is associated with the
8 client public key, sending a key identifier for the client public key from the server
9 to the client, wherein the key identifier allows the client to determine whether the
10 client possesses the client public key.

1 12. The computer-readable storage medium of claim 10,
2 wherein the request for approval includes key reconstitution information
3 that allows the client to decrypt to an encrypted client private key at the client if
4 the client forgets a passphrase for decrypting the encrypted client private key; and
5 wherein the method further comprises storing the key reconstitution
6 information in the database.

1 13. The computer-readable storage medium of claim 10, wherein the
2 method further comprises:
3 decrypting the request for approval at the server using a server private key,
4 the request for approval having been encrypted with a corresponding server public
5 key by the client; and
6 using the client public key to verify that the request for approval is signed
7 by a corresponding client private key.

1 if an affirmative response to the verification request is not received,
2 removing the client public key from the database.

1 19. An apparatus that facilitates managing public keys through a server
2 that stores associations between public keys and email addresses, the apparatus
3 comprising:

4 a receiving mechanism at the server that is configured to receive a first
5 message from a client, the first message containing a request for approval of a
6 client public key along with the client public key;

7 a sending mechanism that is configured to send a second message to the
8 client, the second message containing a request for identity confirmation that
9 includes the client public key; and

10 a database located at the server;

11 a storing mechanism coupled to the database, wherein if the receiving
12 mechanism receives a third message from the client containing an affirmative
13 response to the request for identity confirmation, the storing mechanism is
14 configured to store an association between a client email address and the client
15 public key in a database, so that other clients can look up the client public key in
16 the database.

1 20. The apparatus of claim 19, further comprising a key lookup
2 mechanism that is configured to:

3 receive a communication from a second client at the server, the
4 communication including the client email address;

5 perform a lookup in the database based on the client email address to
6 determine if the client email address is associated with the client public key; and
7 to

8 send a key identifier for the client public key from the server to the client,
9 if the lookup indicates that the client email address is associated with the client
10 public key, wherein the key identifier allows the client to determine whether the
11 client possesses the client public key.

1 21. The apparatus of claim 19,
2 wherein the request for approval includes key reconstitution information
3 that allows the client to decrypt to an encrypted client private key at the client if
4 the client forgets a passphrase for decrypting the encrypted client private key; and
5 wherein the storing mechanism is additionally configured to store the key
6 reconstitution information in the database.

1 22. The apparatus of claim 19, further comprising:
2 a decryption mechanism that is configured to decrypt the request for
3 approval at the server using a server private key, the request for approval having
4 been encrypted with a corresponding server public key by the client; and
5 a verification mechanism that is configured to use the client public key to
6 verify that the request for approval is signed by a corresponding client private key.

1 23. The apparatus of claim 19, further comprising a lookup mechanism
2 that is configured to determine if the database already contains a prior client
3 public key associated with the client email address;
4 wherein if the database already contains the prior client public key, the
5 sending mechanism is additionally configured to include the prior client public
6 key in the request for identity confirmation sent to the client, so that the client can
7 indicate that the server should replace the prior client public key with the client
8 public key.

1 24. The apparatus of claim 19, further comprising a key removal
2 mechanism that is configured to:
3 receive a request at the server to remove the client public key from the
4 database; and to
5 remove the client public key from the database, if the request is signed
6 with a corresponding client private key.

1 25. The apparatus of claim 19, wherein the database contains at most
2 one key for each email address.

1 26. The apparatus of claim 19, wherein the database contains at most
2 one email address for each key.

1 27. The apparatus of claim 19, further comprising a key removal
2 mechanism that is configured to:
3 send a verification request from the server to the client email address
4 asking if the client public key remains valid; and to
5 remove the client public key from the database, if an affirmative response
6 to the verification request is not received.

METHOD AND APPARATUS FOR MANAGING PUBLIC KEYS THROUGH A SERVER

ABSTRACT

One embodiment of the present invention provides a system for managing public keys through a server that stores associations between public keys and email addresses. This system operates by receiving a first message from a client containing a request for approval of a client public key along with the client public key. In response to this request for approval, the system sends a second message to the client containing a request for identity confirmation that includes the client public key. If a third message is received from the client containing an affirmative response to the request for identity confirmation, the system stores an association between a client email address and the client public key in a database. This allows other clients to look up the client public key in the database.

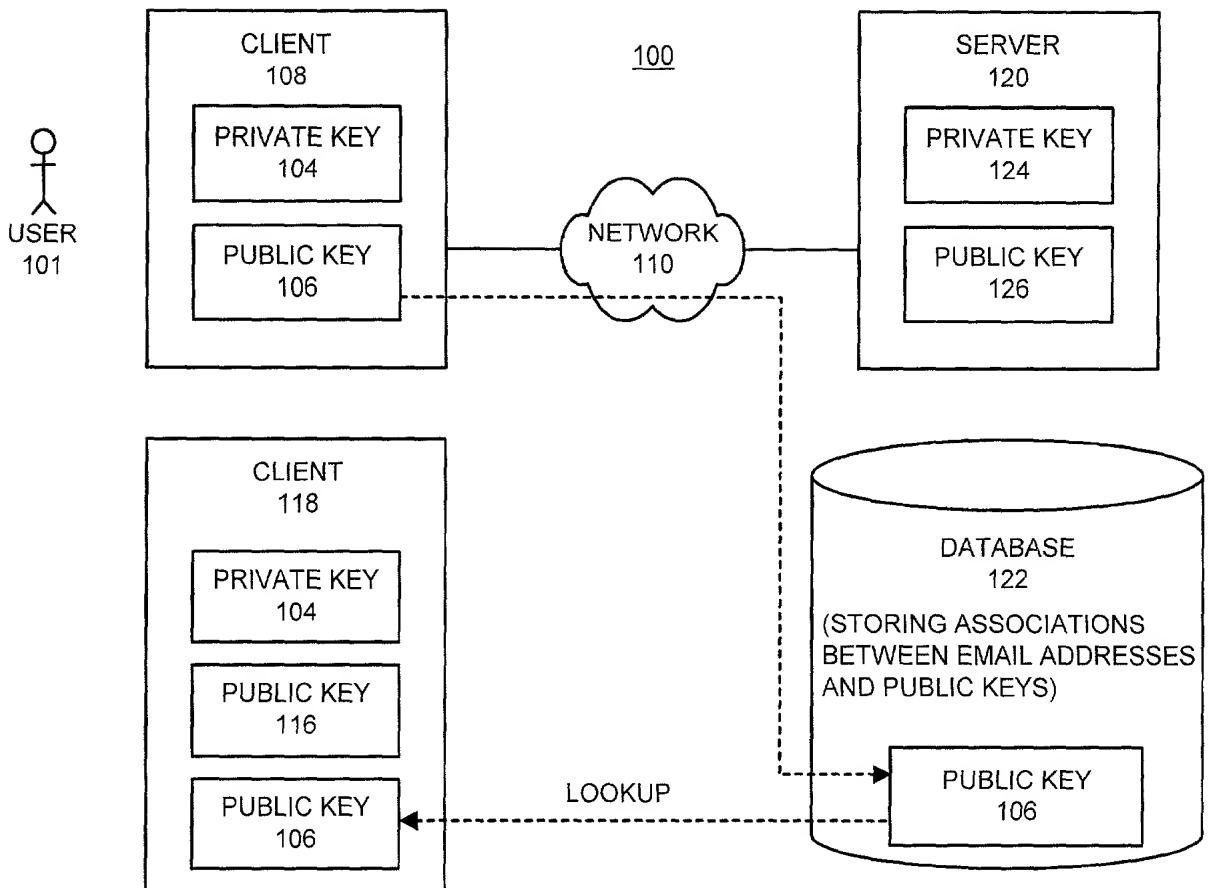


FIG. 1

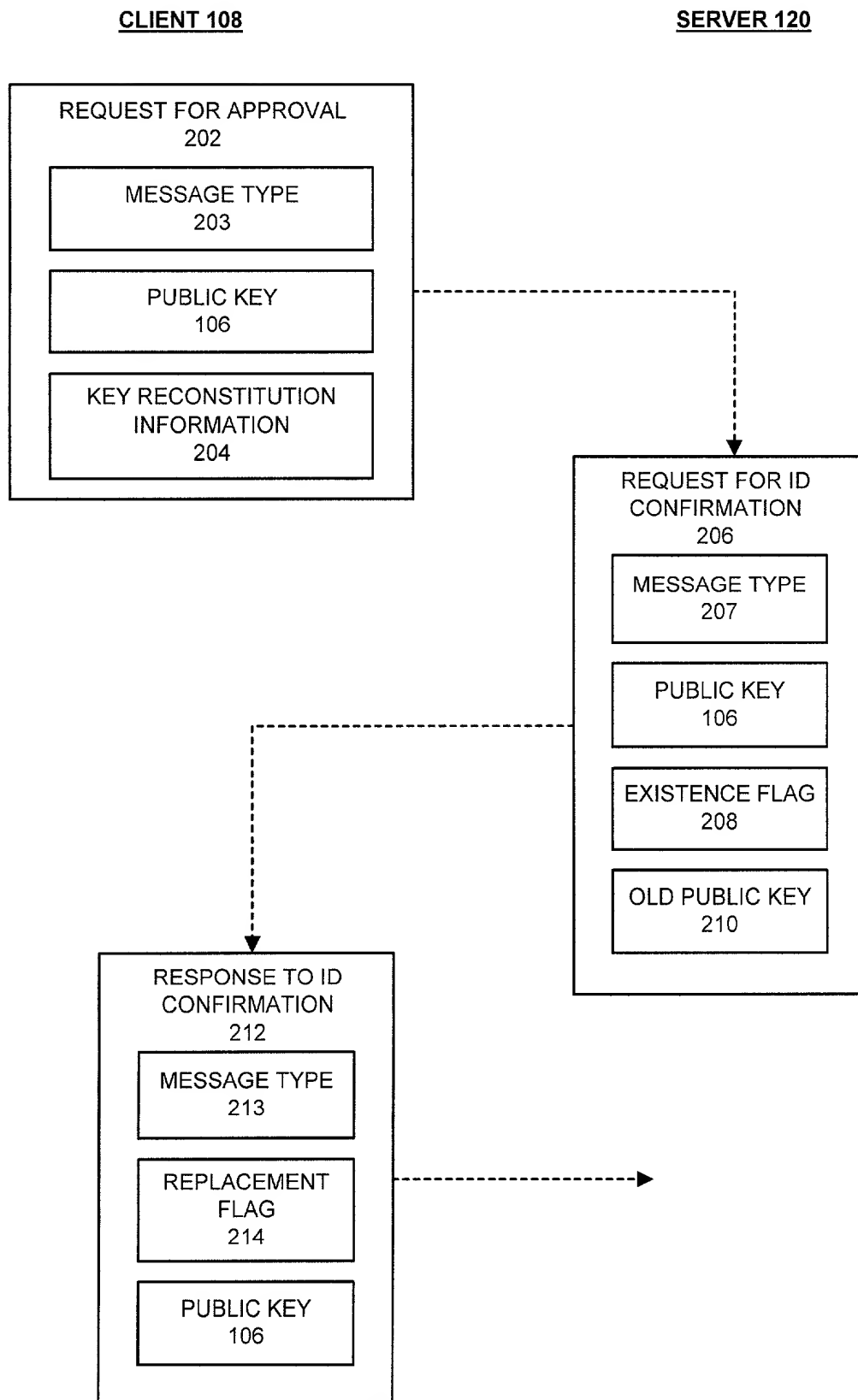


FIG. 2

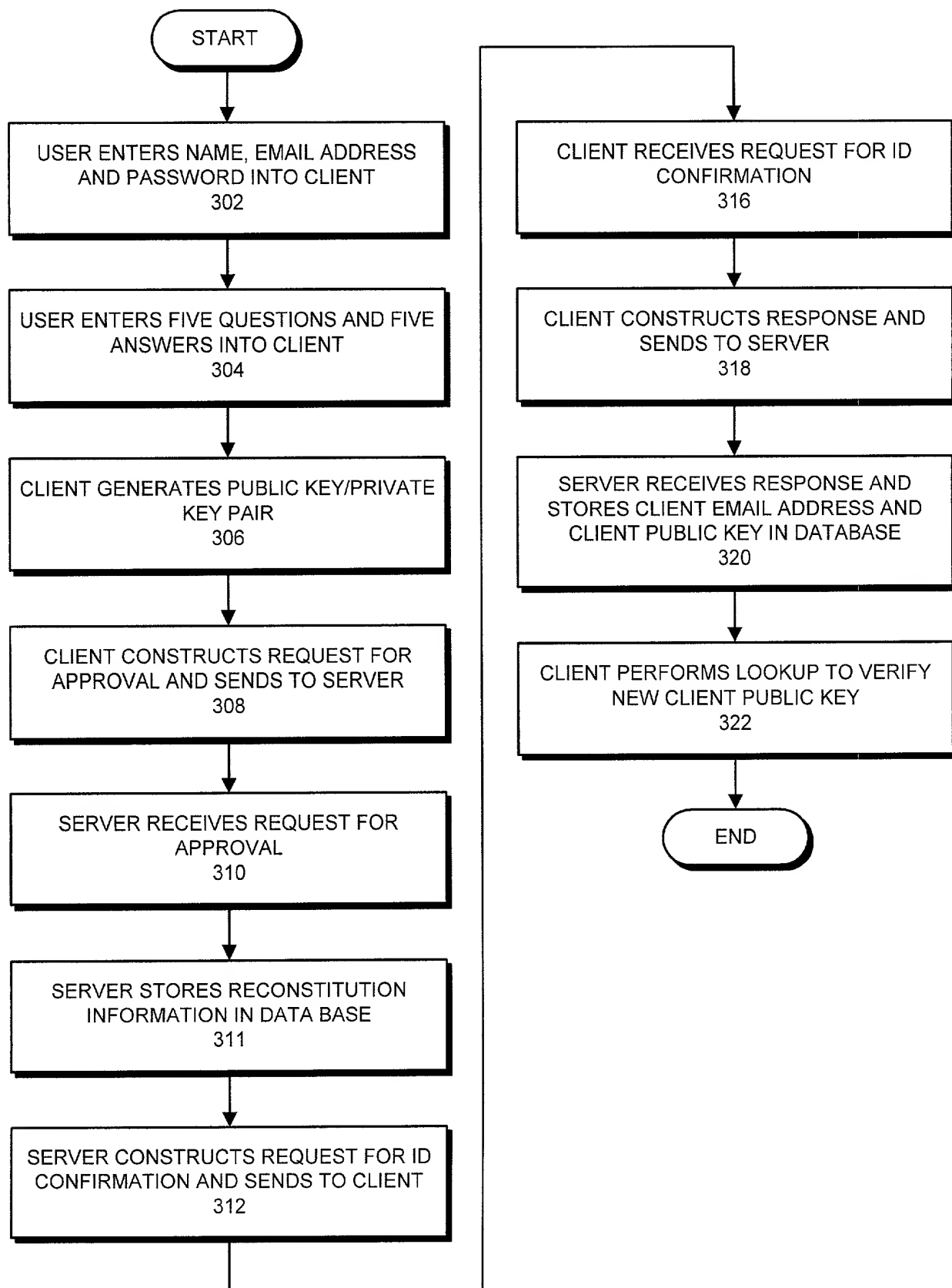


FIG. 3

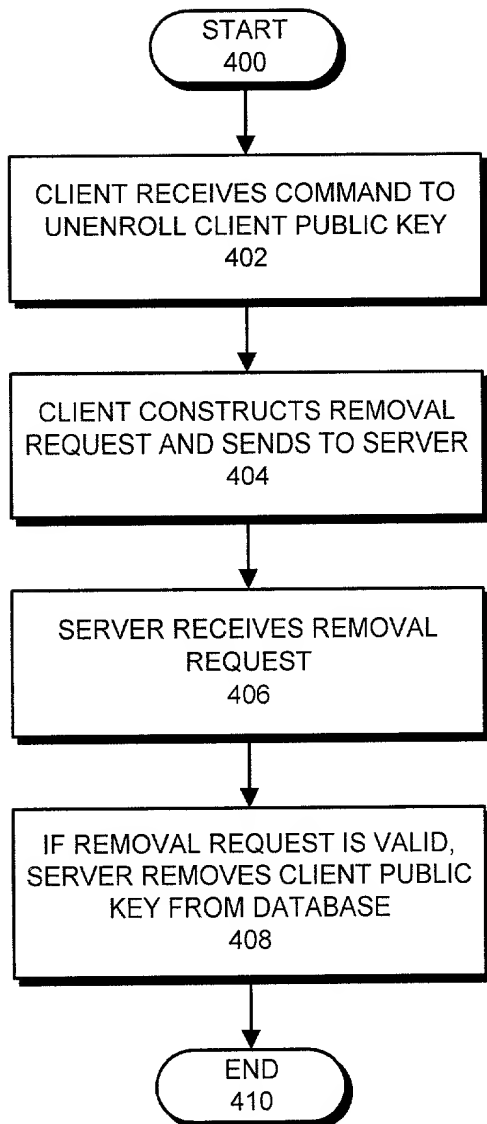


FIG. 4

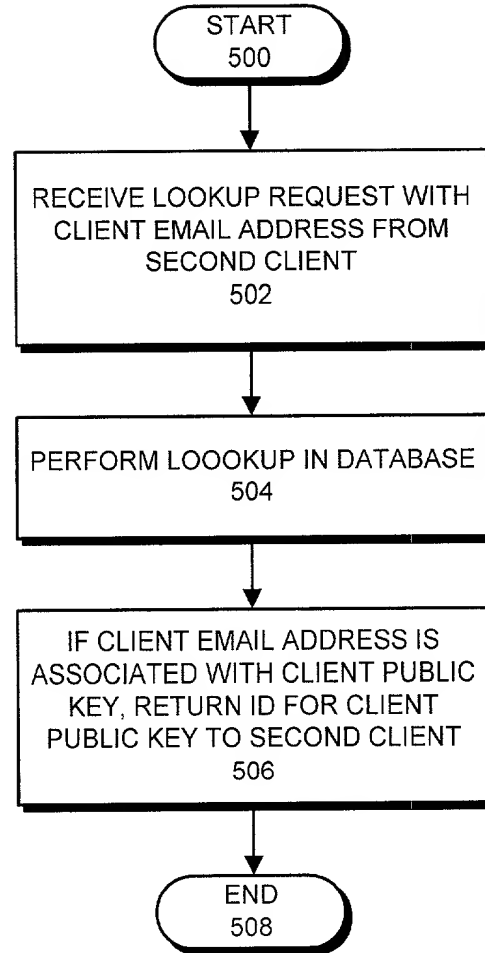


FIG. 5

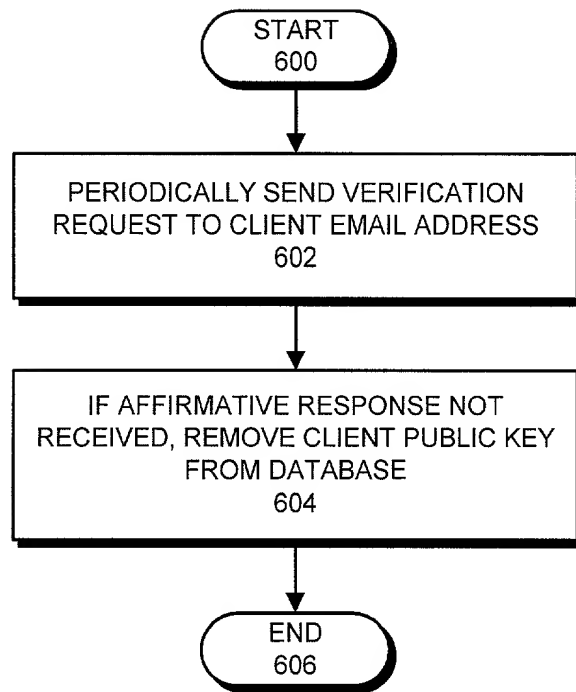


FIG. 6

COMBINED DECLARATION AND POWER OF ATTORNEY

As a below-named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below by my name;

I believe I am the original, first and sole inventor, if only one name is listed below, or an original, first and joint inventor if multiple names are listed below, of the subject matter which is claimed and for which a patent is sought on the invention entitled:

METHOD AND APPARATUS FOR MANAGING PUBLIC KEYS THROUGH A SERVER

for which a patent application:

☒ is attached hereto.

☐ was filed in the United States on _ as Application No. _;

☐ with amendment(s) filed on _____ (if applicable).

I hereby state that I have reviewed and understand the contents of the application identified above, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information known to me to be material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56, which states in relevant part:

Each individual associated with the filing and prosecution of a patent application has a duty of candor and good faith in dealing with the Office, which includes a duty to disclose to the Office all information known to that individual to be material to patentability as defined in this section... The duty to disclose all information known to be material to patentability is deemed to be satisfied if all information known to be material to patentability of any claim issued in a patent was cited by the Office or submitted to the Office.

I hereby claim foreign priority benefits under Title 35, United States Code, §119(a)-(d), of any foreign application(s) for patent or inventor's certificate as indicated below and have also identified below any foreign application for patent or inventor's certificate on this invention having a filing date before that of the application on which priority is claimed:

EARLIEST FOREIGN APPLICATION(S), IF ANY, FILED PRIOR TO THE FILING DATE OF THE APPLICATION			
APPLICATION NUMBER	COUNTRY	DATE OF FILING (Day, Month, Year)	PRIORITY CLAIMED
			YES <input type="checkbox"/> NO <input type="checkbox"/>

I hereby claim the benefit under Title 35, United States Code, §119(e), of any United States provisional application(s) listed below:

APPLICATION NUMBER	DATE OF FILING
60/230,235	9/1/2000

I hereby claim the benefit under Title 35, United States Code, §120, of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose information that is material to patentability as defined in Title 37, Code of Federal Regulations, §1.56, which became available between the filing date of the prior application and the national or PCT international filing date of this application:

APPLICATION NUMBER	DATE OF FILING	STATUS		
		PATENTED	PENDING	ABANDONED

I hereby appoint Daniel E. Vaughan (Reg. No. 42,199) and A. Richard Park (Reg. No. 41,241) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith, and to file, prosecute and transact all business in connection with international applications directed to said invention.

Address correspondence to:
Park & Vaughan LLP
508 Second Street, Suite 201
Davis, CA 95616



22835
 PATENT TRADEMARK OFFICE

Direct telephone calls to:
 A. Richard Park
 (530) 759-1661

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Title 18, United States Code, §1001, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

1	Name and Citizenship	William F. Price III	United States of America
	Residence Address	35 Alma Ct., Los Altos, CA 94022	
	Postal Address (if different from Residence)		
	Signature and Date	William F. Price III	Date 11/21/00
2	Name and Citizenship		
	Residence Address		
	Postal Address (if different from Residence)		
	Signature and Date		Date
3	Name and Citizenship		
	Residence Address		
	Postal Address (if different from Residence)		
	Signature and Date		Date
4	Name and Citizenship		
	Residence Address		
	Postal Address (if different from Residence)		
	Signature and Date		Date
5	Name and Citizenship		
	Residence Address		
	Postal Address (if different from Residence)		
	Signature and Date		Date

Additional inventor name(s) and signature(s) attached?: YES ☐ NO ☒